REMARKS

This is intended as a full and complete response to the Final Office Action dated May 9, 2005, having a shortened statutory period for response set to expire on August 9, 2005. Reconsideration and allowance of the claims pending in the application is requested for reasons discussed below.

In this office action, claims 5-13 were provisionally rejected over claims of co-pending application 10/609,967. Consideration of this rejection is respectfully requested to be deferred until a final decision is reached on the claims of this application and the claims of the cited application.

Claims 1-20 are rejected under 35 U.S.C. §103 as unpatentable over *Taylor et al.*, U.S. Patent 6,630,935, in view of *Rentschler et al.*, U.S. Patent 5,969,726. This rejection is respectfully traversed. The claims will be discussed below in three separate groups.

Claims 1-4

Claims 1-4 are rejected over *Taylor* and *Rentschler* with special reference being made to *Rentschler* as teaching a global state value which, according to the Examiner, avoids providing the graphics processor with an excessive amount of data. This is respectfully traversed. *Rentschler* does not teach the type of global state value disclosed herein at paragraphs 59 and 60 and claimed in the claims 1-4.

Claims 1-4 recite that the allocation of each of the sample types is determined and limited by a sample portion global state value which determines, for each sample type, the number of samples which may be processed by one or more of the threads of the multithreaded processing unit. Thus, each of the sample types, as recited, has an associated sample portion global state value. Further, as recited for example in claim 4, all sample portion global state values may be determined dynamically, and changed in the course of processing of data. In contrast, the global state value taught by *Rentschler* is truly a global value, which is applied to all or substantially all vertices and rarely changes during the rendering of a model or object (column 13:30-45). It is not applied differently to each sample type. Therefore, the global state value of *Rentschler*

does not anticipate or make obvious the recitation of claims 1-4, and reconsideration and allowance of these claims is requested.

Claim 5

Claim 5 recites a graphics processor, including a thread control unit with an output buffer, a thread storage resource which includes a write flag indicating which of the threads will write data to one of the output pixel positions. The Examiner, while citing a teaching in *Taylor* of an output buffer as claimed, provides no citation to a write flag in a thread storage resource as claimed. Further, the review of the reference does not find any such teaching. Therefore, reconsideration and allowance of this claim is requested.

Claim 6-20

These claims are rejected over the teachings of *Taylor*. It is respectfully submitted that this is a faulty interpretation of the teachings of *Taylor* in several respects. For example, the Examiner argues at the bottom of page 4 and the top of page 5 of the office action that *Taylor* demonstrates that execution pipelines are dynamically allocated to simultaneously execute threads. This interpretation is contrary to the clear teachings of *Taylor* beginning at column 31:25 which describe a priorization scheme for using a single computation engine 12, driven by the arbitration module 14, which in turn is fed by the threads 114-130; importantly, the data from threads is processed sequentially. For example, referring specifically to column 31, *Taylor* teaches "thread controller 1 must wait for the latency requirements associated with AOP1 to be satisfied before AOP2 can be submitted for processing; thus thread controller 1 submits AOP1 for execution during cycle C0" (31:47-48). *Taylor* further teaches that "vertex B as received in thread controller 2 is able to submit operation code BOP1 for processing during cycle C1" (31:60-62) and that "AOP2 is not submitted by thread controller 1 until cycle C2" (31:54-55). It is clear from these quotes that the computation engine 12 operates on data from various threads sequentially, i.e. in a sequence of cycles, and that this is the only proper interpretation of the teachings of the *Taylor*.

The Examiner further alleges that *Taylor* teaches determining how many of the threads are assignable to each of the sample types based on a load balance scheme and application allocation priority. This is respectfully traversed. The undersigned has searched the text of the *Taylor* reference, and finds no reference to load balancing. A search for allocation priority only reveals teachings related to Fig. 7 where certain threads have differing priorities for execution, based on which operations have been waiting longest for execution (a difficulty that is clearly apparent when only a single pipeline computation engine is available for the processing). (See 31:24-30.)

No teaching can be located in the *Taylor* patent that infers redistribution of samples among the simultaneous processing capability provided in multiple threads. In contrast, this simultaneous processing is clearly taught in the present application (see, for example, Fig. 3, processing units PCU 375) and claimed herein.

In view of these clear distinctions, all of which have previously been presented in the claims of the application (see claims 1, 5 and 12), reconsideration and allowance of the claims is respectfully submitted.

Having addressed all issues set out in the office action, Applicants respectfully submit that the claims are in condition for allowance and respectfully request that the claims be allowed.

Respectfully submitted,

James A. Sheridan, Reg. No. 25,435
MOSER, PATTERSON & SHERIDAN, L.L.P.
3040 Post Oak Blvd. Suite 1500
Houston, TX 77056
Telephone: (713) 623-4844
Facsimile: (713) 623-4846
Attorney for Applicant(s)